

## An Ontological Framework for Autonomous Systems Modelling

Julita Bermejo-Alonso<sup>\*</sup>, Ricardo Sanz<sup>†</sup>, Manuel Rodríguez<sup>‡</sup> and Carlos Hernández<sup>§</sup>

*Autonomous System Laboratory (ASLab)*

*Universidad Politécnica de Madrid, Madrid, Spain*

<sup>\*</sup> *Email: jbermejo@etsii.upm.es*

<sup>†</sup> *Email: ricardo.sanz@upm.es*

<sup>‡</sup> *Email: manuel.rodriguez@upm.es*

<sup>§</sup> *Email: carlos.hernandez@upm.es*

**Abstract**—Conceptual modelling aims at identifying, and characterising the entities and the relationships of a selected phenomenon in some domain. The obtained conceptual models express the meaning of the concepts used by domain experts, and the relationships between them. An ontology is a formal specification of a common conceptualisation shared by stakeholders and experts in a domain. Ontologies can serve as the semantic support for conceptual modelling, guiding and constraining the intended meaning of the conceptual models. We have followed this approach in our model-based control systems, by developing a domain ontology and an ontology-based methodology to support the conceptual modelling of autonomous systems. The ontology for autonomous systems captures the ontological elements to describe an autonomous system's structure, function, and behaviour. The methodology exploits the ontology to generate the conceptual models for a generic engineering process. Both elements have been used in case studies to obtain conclusions on the suitability of the developed ontology and its application in the model-based engineering of autonomous systems.

**Keywords**-ontology-based engineering; autonomous systems; ontology-driven conceptual modelling.

### I. INTRODUCTION

Our research addresses the development of a universal technology for autonomous systems, in a model-based control paradigm, where the autonomous system uses models as its main knowledge representation supporting the control system's operation. The first stage is to develop these conceptual models based on the design knowledge used by developers to describe and to engineer them. The next step is for the autonomous system itself to use these models, to decide about the actions to be taken. The aim is to make the autonomous system to operate with the same models the engineers use to build it, to be used by cognitive control loops in the autonomous system to increase their autonomy.

We have followed an ontological approach to support autonomous systems conceptual modelling and engineering [1]. The developed framework consists of two intertwined elements: a domain ontology and an ontology-based methodology. The ontology for autonomous systems (OASys) captures the ontological elements to describe an autonomous system's structure, function, and behaviour. The OASys-based methodology exploits OASys to generate conceptual

models for a generic engineering process. Both elements have been used in case studies, as to obtain conclusions on the suitability of the developed ontology and its application in the model-based engineering of autonomous systems.

The paper is structured as follows. Section II describes the Autonomous System (ASys) research programme under development for a technology of autonomous systems. Section III reviews related work on ontologies for autonomous systems. Section IV provides the description of the framework: an ontology for autonomous systems, as well as the methodology which exploits the ontological elements. Section V presents the case studies and their models, explaining how the conceptual models have been developed following the ontology-based methodology. Section VI provides some concluding remarks, as well as suggesting further work to address in next stages of our research.

### II. THE AUTONOMOUS SYSTEM RESEARCH PROGRAMME

The context is the long-term research project ASys, which addresses the development of a technology for the systematic development of autonomous systems. This includes methods for analysing requirements, architectures of autonomy and reusable assets with a cross-domain approach.

During their operation, systems might need to put up with external perturbations, changes from the original specification, or unexpected dynamics not always predicted. Production and automation engineers desire autonomous systems, capable of working on its own. However, this autonomy is considered to be bounded, i.e., the system to be fully autonomous but constrained by the engineers who have developed it. This is, in a sense, a strong point of difference between natural and artificial autonomy. Natural autonomous systems are considered to be truly autonomous systems in the etymological sense of the word (i.e., following their own behavioural laws). On the other side, artificial autonomous systems shall behave autonomously but only to a certain extent, being bounded by externally imposed restrictions (e.g., concerning safety, economics or environmental impact).

The strategical decision in the ASys project is the consideration of all the domain of autonomy, i.e., to cater for any

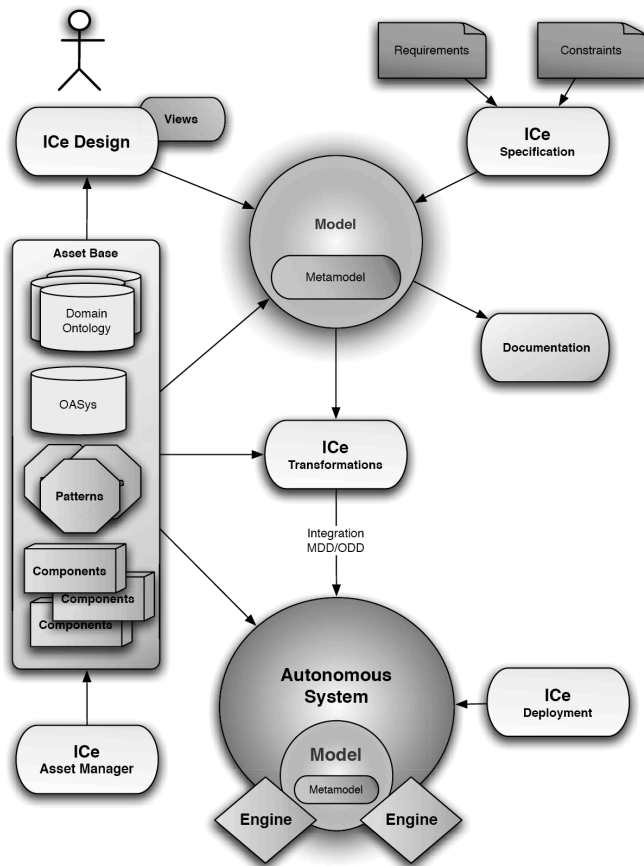


Figure 1. Elements in the ASys Research Programme

autonomous system regardless of its particular application. This implies a wide range of (autonomous) systems to be considered in the research, from robot-based applications to continuous processes or pure software systems. While this movement towards generality may imply a substantial reduction of powerfulness of the developments (abstraction may convey vacuity) the progressive domain focalisation [2] method will enable the derivation of progressively domain-focused assets that are, at the same time, compliant with the abstractions and capable of providing functional value. The strategy followed to increase a system's autonomy will be by exploiting a particular class of patterns: *cognitive control loops*, which are control loops based on knowledge [3].

The research programme considers different elements to materialise the former ideas (Figure 1): an architecture-centric design approach, a methodology to engineer autonomous systems based on models, and an asset base of modular elements to fill in the roles specified in the architectural patterns.

Our engineering process covers from the initial specifications and knowledge, to the final product, i.e., the autonomous system. The first stage of the research programme focuses on ontologies, as a common conceptualisation to

describe domain knowledge. Both a survey of existing domain ontologies and the development of an ontology for the domain of autonomous systems (OASys) [4] are addressed. One of the central goals is to produce a methodology to exploit these ontologies to generate models based on the knowledge they contain.

A cornerstone of the ASys programme is the use of design patterns as the core vehicle for reusable architecture exploitation [5]. Design patterns present solutions to recurring design problems in a certain context. ASys patterns could be classified in two categories: *architectural patterns*, that express the structural organization of an autonomous system, i.e., they realise the architecture, and *domain patterns*, that describe a mechanism to solve a concrete but recurring problem in a particular context, in a similar way to Buschmann's categorisation into architectural patterns and design patterns [6]. Patterns will not be used in ASys independently from the OASys ontology. Domain patterns will describe interactions of the system's components and with the environment, by using the conceptualisation of the ontology, that represent design solutions so that the behaviour of the system fulfills the engineering requirements. They will model the intended - designed - system's dynamics with its environment. On the other hand, architectural patterns will do the same for the internal system's organization and dynamics, describing them in terms of interactions in between the ontological elements that conceptualize the system itself. Thus all system patterns will not only be specified departing from the OASys concepts, but eventually will become part of the ontology itself, modelling the relations and interactions between them as designed by the engineers.

Models constitute the core for our autonomous systems research programme. The type and use of models to be specifically developed for the autonomous system are initially considered. User and designer requirements, and constraints imposed by the system itself will guide the development of the models. The ASys Model Development Methodologies will address this model characterisation and development. The next stage is to extract from the built models a particular view of interest for the autonomous system. Unified functional and structural views are considered critical for our research as they provide knowledge about the intentions and the behaviours of the autonomous system.

The obtained models will be exploited by means of commercial application engines or customised model execution modules. Considering the metacognitive needs of autonomous systems, metamodels are addressed in the research. Progressive domain focalisation will be used to address the different levels of abstraction between metamodels and models. As an additional product documentation about the built models for their update, exchange, and query will be obtained.

The ASys programme approach is conceptual and architecture-centric. The suitability of extant control and

cognitive control architectures shall be determined, in terms of how they match the ASys research ideas and developed products (ontologies, models, views, engines). Possible adaptations and extensions of the analysed architectures shall also be considered.

The definition and the consolidation of architectural patterns that capture ways of organising components in functional subsystems will be critical. As a generalisation strategy, the different elements considered in the ASys research programme will be assessed in different testbeds.

### III. RELATED WORK

An ontology is a formal specification of a common conceptualisation shared by stakeholders and experts in a domain [7] [8]. Two dimensions can be considered in any ontology: the *pragmatic dimension* to express its application to a certain domain with an intended use, developed following a specific methodology or design method; and the *semantic dimension* as a representation mechanism that structures, organises and formalises a particular content, according to a level of granularity.

An ontology contains classes, relationships, instances and axioms. Classes correspond to entities in the domain under analysis. Relationships relate such entities. Instances are the actual objects that are in the domain. Axioms constrain the use of all former elements.

Focusing on autonomous systems, ontologies are used as a knowledge representation mechanism, in terms of a specification of a conceptualisation. Hence, the ontology contains the concepts to be manipulated by the actors of the autonomous system. The ontological elements are defined based on a computational language, such as OWL or UML.

A first example of ontologies developed for autonomous systems relates to their use for mobile robots. The underlying idea is to conceptualise the different entities taking part in the operation of a mobile robot. Ontologies describe the environment, as a repository of the objects in it [9] or the location the robots are moving in [10] [11]. Mobile robots also need to construct and manipulate representations of the surrounding environment built up by means of sensors, where ontologies represent spatial knowledge [12] [13] [14].

Mobile robots are usually faced with real-time and complex tasks which might require extremely large knowledge to be stored and accessed. Ontologies help to structure this knowledge and its different levels of abstraction [15], to describe task-oriented concepts [16], to act as metaknowledge for learning methods and heuristics [17] or to define concepts related to actions, actors and policies to constraint behaviour [18].

Additionally, ontologies have been used for agent-based systems, where the stress has been on using ontologies for knowledge sharing and exchange among the different agents in the system [19]. Common, global or shared ontologies are used to overcome the semantic heterogeneity among agents.

Commitments to the shared ontology permit the agents to interoperate and cooperate while maintaining their autonomy [20] [21] [22] [23] [24].

Recently, ontologies have been used within autonomic computing developments, as a knowledge representation mechanism to provide support for information exchange and integration. Autonomic systems require knowledge from different sources to be represented in a common way. The shared conceptualisation allows to reduce structural and semantic heterogeneity, which appears when the autonomic system handles data within different schemes, contents or intended meanings. Moreover, having ontology axioms and rules to verify the model play a major role, when autonomic systems are faced to a high heterogeneity of data and semantics [25] [26] [27]. As for other autonomous systems, ontologies have been developed to describe the autonomic system [28] or its environment [29].

Autonomous systems have thus benefited of using ontologies for their design and operation [30] [31] :

- Ontologies clarify the structure of knowledge: performing an ontological analysis of a domain allows to define an effective vocabulary, assumptions and the underlying conceptualization. The analysis also allows to separate domain knowledge from operational or problem-solving one.
- Ontologies help in knowledge scalability: knowledge analysis can result in large knowledge bases. Ontologies help to encode and manage them in a scalable way.
- Ontologies allow knowledge sharing and reuse: by associating terms with concepts and relationships in the ontology as well as a syntax for encoding knowledge in them, ontologies allow further users to share and reuse such knowledge.
- Ontologies increase the robustness: ontological relationships and commitments can be used to reason about novel or unforeseen events in the domain.
- Ontologies provide a foundation for interoperability among heterogeneous agents and system's elements.

Hence, ontologies provide a common conceptualisation that can be shared by all those involved in an engineering development process. They also procure a good mean to analyse the knowledge domain, allowing the separation of descriptive and problem-solving knowledge. They can be as generic as needed allowing its reuse and easy extension. Ontologies can serve as the semantic support for conceptual modelling, guiding and constraining the intended meaning of the conceptual models of the autonomous system.

### IV. THE OASYS FRAMEWORK

The OASys Framework captures and exploits the concepts to support the description and the engineering process of any autonomous system. This has been done by developing two different elements:

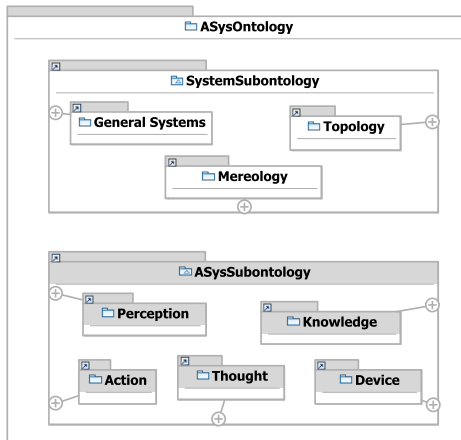


Figure 2. ASys Ontology structure

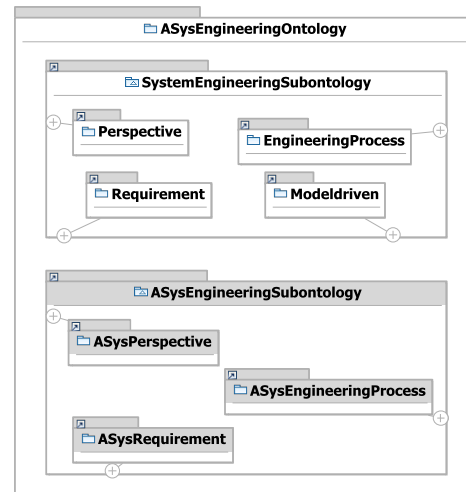


Figure 3. ASys Engineering Ontology structure

- An autonomous systems domain ontology (OASys): The term autonomous systems covers too broad a domain, being applied to a wide range of systems. Finding common features and elements shared by different autonomous systems will aid for their description and engineering. Hence, a domain ontology describing concepts, relationships and axioms related to the autonomous systems domain, as a representation-based mechanism built on UML (a tool suitable to represent ontologies as well as to be used for model-driven engineering to develop the final software assets). The aim is to provide generic building elements to cater from generic autonomous systems to specific applications.
- An OASy-based Engineering Methodology: Once the domain ontology is described, it serves as the basis for an ontology-based engineering process for autonomous systems, whose structure aims at being re-usable and scalable.

Software-intensive systems are usually developed in an ad-hoc engineering process, i.e., specifically built to fulfill specific requirements of a project or application, without general guidelines which will allow a further partial or full re-use. Furthermore, a myriad of different users (either human or artificial) interact in the engineering process, generally without sharing a common conceptualization of the problem. The development process is usually prone to misunderstandings, as well as time- and effort-consuming. A common conceptualisation of relevant concepts and development techniques will be useful to aid in the requirements specification and development phases of an autonomous system engineering process.

#### A. The Ontology for Autonomous Systems: OASys

OASys has been developed to describe the domain of autonomous systems, as software and semantic support for the conceptual modelling of autonomous system's description

and engineering. The pragmatic dimension of the ontology, i.e., its application to the domain of autonomous systems, has been addressed considering two different ontologies as part of it: the ASys Ontology to conceptualise the autonomous system's description, and the ASys Engineering Ontology to do likewise with the autonomous system's engineering process.

For its development, we have followed the METHONTOLOGY methodology [32] that provides the guidelines and steps for ontological engineering, from knowledge acquisition to ontology evaluation.

As starting point, the *knowledge acquisition phase* where documents and ontologies have been reviewed. Documents have been analysed to come up with existing terminology and definitions for the different domains, subdomains, applications and aspects considered in the ontology's structure. The sources included articles, technical reports describing body of knowledge, and books. As underlying focus, the research and ideas developed in the ASys research programme. Existing well-established glossaries and ontologies have also been assessed to be integrated in the ontology.

Next, the *development activities* to obtain a prototype of the ontology which evolves as new versions are considered. These activities encompass, among others, the specification, the conceptualisation, and the formalisation of the ontology:

- The specification activity aims at answering questions such as which domain is considered, which use is given to the ontology, and who will use it.
- The conceptualisation activity organises and structures the knowledge acquired using external representations that are independent of the knowledge representation and implementation paradigms in which the ontology will be formalised and implemented afterwards.
- The formalisation activity has as goal to formalise the

Table I  
SYSTEM SUBONTOLOGY PACKAGES

| Package         | Purpose  |
|-----------------|--|
| General Systems | To provide concepts to characterise any kind of system's structure, function and behaviour |
| Mereology       | To gather general concepts for whole-part relationships                                    |
| Topology        | To collect general concepts for topological connections                                    |

Table II  
ASYS SUBONTOLOGY PACKAGES

| Package    | Purpose   |
|------------|---|
| Perception | To define concepts to describe the perceptive and sensing process in an autonomous system                                 |
| Knowledge  | To specify concepts to describe the different kinds of knowledge (models, ontologies, goals) used by an autonomous system |
| Thought    | To characterise concepts to describe the goal-oriented processes in an autonomous system                                  |
| Action     | To collect concepts about the operations carried out and performing actors in an autonomous system                        |
| Device     | To gather concepts to describe the autonomous system's devices  |

conceptual model. This has been made using UML [33] to implement the ontological elements in the ontology.

To consider the semantic dimension of the ontology, the contents of the ASys and ASys Engineering Ontologies have been organised and structured into subontologies and packages. Subontologies are used to organise the ontological elements into generic knowledge to domain-specific ones. Each subontology contains different packages that gather aspect-related concepts and relationships. Both subontologies and packages have been designed to allow future extensions and updates to OASys.

The *ASys Ontology* contains the concepts, relations, attributes and axioms to characterise an autonomous system, organised in two subontologies (Figure 2):

- The System Subontology contains the elements necessary to define any systems structure, behaviour and function, based on general and well-established theories, which consists of different packages (Table I).
- The ASys Subontology elements specialise the System Subontology concepts and relationships for an autonomous system. The subontology is organised in different packages which address the different aspects and functionalities in an autonomous system: the perception, thought, and action activities based on knowledge that an autonomous system performs (Table II).

The *ASys Engineering Ontology* collects the ontological elements to describe and support the construction process of an autonomous system. It comprises two subontologies to address at a different level of abstraction the ontological

Table III  
SYSTEM ENGINEERING SUBONTOLOGY PACKAGES

| Package             | Purpose  |
|---------------------|--|
| Requirement         | To gather concepts to describe systems requirements                |
| Perspective         | To provide concepts to describe different perspectives in a system |
| Engineering Process | To define concepts to describe the engineering process             |
| Model-driven        | To collect concepts to describe model-driven engineering           |

Table IV  
ASYS ENGINEERING SUBONTOLOGY PACKAGES

| Package                  | Purpose  |
|--------------------------|--|
| ASys Requirement         | To specify concepts to describe the requirements in an autonomous system         |
| ASys Perspective         | To gather concepts to describe an autonomous system from different aspects       |
| ASys Engineering Process | To provide concepts to describe the construction process of an autonomous system |

elements to describe the autonomous system's engineering development process (Figure 3).

- The System Engineering Subontology gathers concepts and relationships related to process and software system's engineering. It is based on different metamodels, specifications and glossaries which have been long used for software-based developments. The subontology contains different packages to address different aspects to consider, as needed, within a system's development: requirements, engineering process, perspective, and model driven (Table III).
- The ASys System Engineering Subontology contains the specialisation of the System Engineering Ontology contents, as well as additional ontological elements to describe a concrete autonomous system's engineering process, organised as different packages (Table IV). To mention that this subontology does not yet include an ASys Model-driven package, since the specific model-driven development for autonomous systems in the ASys research project will be later specified.

### B. The OASys-based Engineering Methodology

The OASys-based Engineering Methodology is a generic ontology-based, autonomous systems engineering method based on OASys ontological elements. The methodology focuses on the description on how to carry out the autonomous system generic engineering process, having as guideline the ontological elements in the System Engineering and ASys System Engineering Ontologies. Being OASys-based, the methodology considers the ontological elements required in

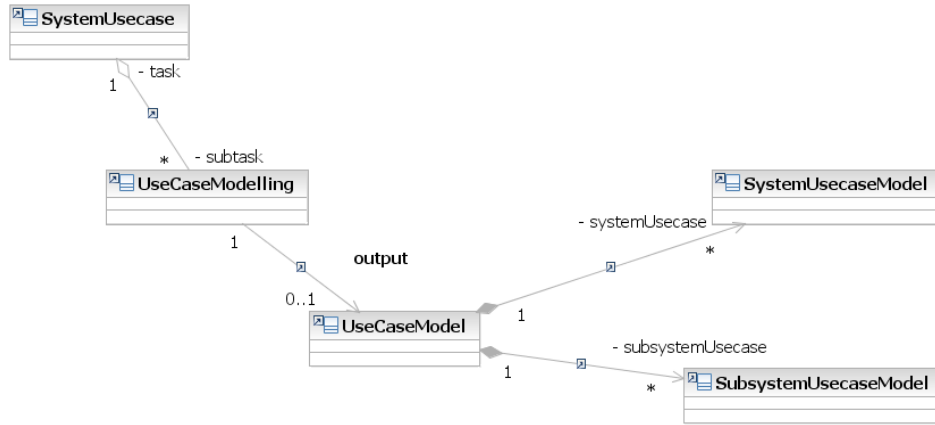


Figure 4. The Use Case Modelling subtask



Figure 5. The Use Case Detailing subtask

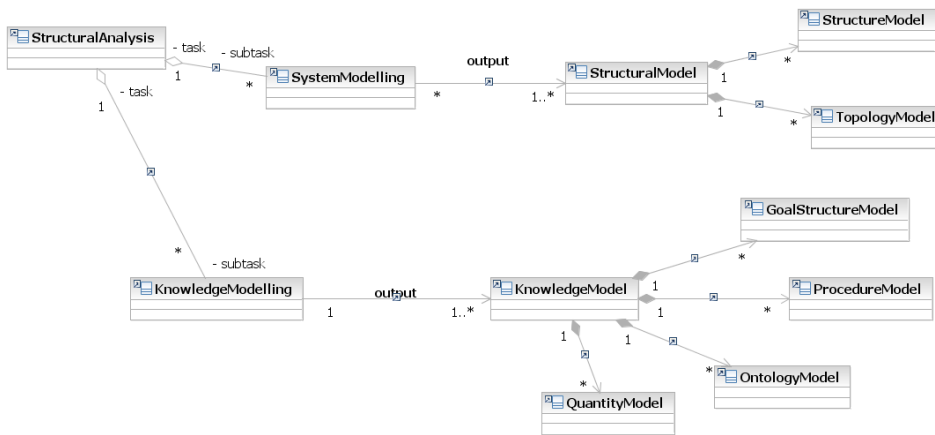


Figure 6. The Structural Analysis task

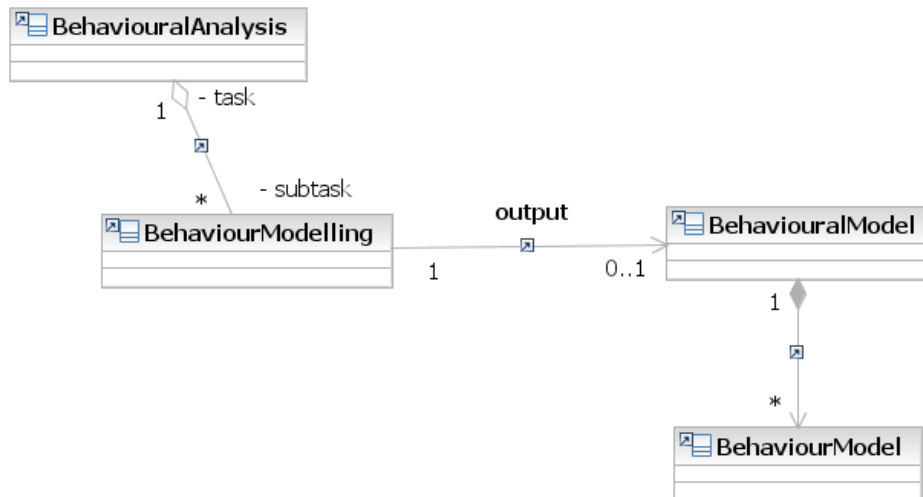


Figure 7. The Behavioural Analysis task

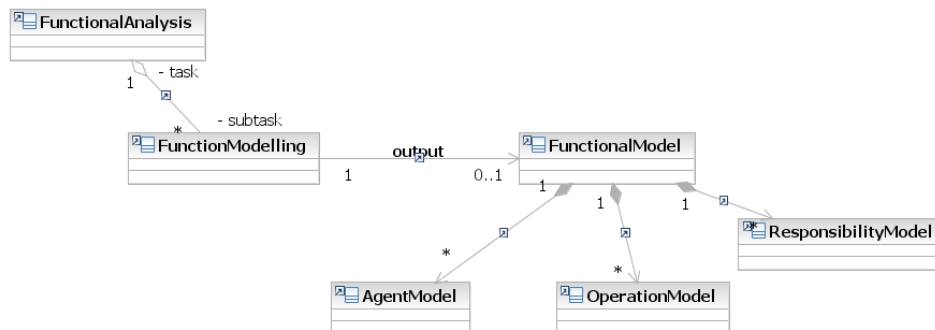


Figure 8. The Functional Analysis task

the different tasks, by specifying the OASys packages to be used.

The engineering methodology tries to transfer state-of-the-art systems engineering methods to the lame engineering practices in cognitive autonomous systems construction. The methodology consists of two main phases: ASys Requirement to identify the autonomous system requirements, and ASys Analysis to consider the autonomous system analysis of its structure, behaviour and function.

The *ASys Requirement* phase identifies and elicits stakeholders' requirements for the system, to characterise the processes and the system. Traditional requirements engineering techniques are used to specify the requirements. The System Use Case task obtains the system and subsystem's use cases models through the Use Case Modelling subtask (Figure 4) and the Use Case Detailing subtask (Figure 5). All the elements shown in these UML class diagrams are defined as ontological elements in the ASys Engineering Ontology.

The different models obtained as workproducts of this phase are constructed using the ontological elements defined

in the Requirement and ASys Requirement packages, as it is exemplified in Section V.

The *ASys Analysis* phase describes the autonomous system from different viewpoints, paying attention to the structural, behavioural and functional features in the autonomous system. The concepts and relationships considered to describe the tasks, subtasks and workproducts are defined as ontological elements in the ASys Engineering Ontology.

The Structural Analysis task (Figure 6) considers the system from a structural viewpoint, consisting of different modelling subtasks to analyse the system's subsystems and elements, variables, algorithms, and ontologies. The main work products obtained are the Structural and the Knowledge Models. The Structural Model focuses on modelling the mereotopological relations among the subsystems and elements in the autonomous system, by refining the General Systems, Mereology, and Topology packages. The Knowledge Model considers the different kinds of knowledge for an autonomous system, specialising the Knowledge Package concepts into different models.

The Behavioural Analysis task (Figure 7) targets the system from a behavioural viewpoint, obtaining a Behavioural Model through a Behaviour Modelling subtask, which captures autonomous system behaviour. The Behavioural Model is obtained by specialising the ontological elements in the General Systems Package.

The Functional Analysis task (Figure 8) analyses the system from a functional viewpoint, obtaining a Functional Model that specialises the concepts from the Perception, Thought and Action packages. The Functional Model consists of different models: the Agent Model identifies the different actors in the autonomous system, as defined in the Action Package; the Operation Model captures the operations performed, as specialisation of the Operation taxonomy in the Action Package and the Perception ones defined in the Perception Package; and the Responsibility Model specifies the distribution of responsibilities among the different actors, identifying their responsibilities, possible inputs, outputs and resources to perform an operation.

## V. TESTBED APPLICATIONS

OASys and its related methodology have been applied in the description and engineering of two testbeds considered within the ASys project: the Robot Control Testbed (RCT), and the Process Control Testbed (PCT). The OASys-based Engineering Methodology has been applied to the testbeds during their analysis. Different tasks have been carried out and several work products have been obtained. The next sections exemplify some of the conceptual models obtained for the testbeds. The terms in *italics* refer to those ontological elements defined in the different Subontologies and Packages of OASys. Relationships between concepts do not follow this format, for the sake of simplicity.

### A. The Robot Control Testbed

The Robot Control Testbed (RCT) is a collection of mobile robot systems, with a wide range of implementations and capabilities (from conventional Simultaneous Localisation And Mapping (SLAM) based mobile robots to virtual robots inspired in rat brain neuroscience). The research aim is to develop a general, customisable autonomous robot architecture providing the system with the necessary cognitive capabilities and robustness to perform complex tasks in uncertain environments.

The current robotic application, Higgs, consists of different interrelated systems to provide the desired capabilities for autonomous navigation (Figure 9). Three different sub-systems, composed of several elements, can be considered: the base platform, the onboard systems, and the supporting systems.

The base platform is a mobile robot ActivMedia Pioneer 2-AT8. It is a robust platform, specially designed for outdoor applications. The platform includes all the necessary elements to implement a control and navigating system.



Figure 9. Base platform with onboard systems (Higgs)

Several element have been added to the base platform as onboard systems. Hardware systems have been attached to the platform to expand the range of functionalities, such as an onboard computer, lasers, cameras and a GPS. Moreover, software modules have been developed to provide the mobile robot with further capabilities such as synthetic emotions, communication, surface recognition and real-time processes.

Additionally, other supporting systems have been included to complete the testbed, such as a wireless network, servers and remote controllers to aid in the control operations of the mobile robot.

### B. RCT Conceptual Modelling

This section describes the conceptual models obtained for the current robotic-based system, as a result of applying the OASys-based Engineering Methodology for the RCT *Requirement Phase*. In this *Phase*, the *Requirements* are identified, considering the *RequirementViewpoint* concept defined in the Requirement Package. *Requirements* are *organised* by means of *UseCases* as defined in the Requirement Package. *UseCases* are modelled in the *Use Case Modelling Subtask* as part of the *System UseCase Task*, as defined in the ASys Engineering Process Package.

To analyse the *UseCases*, the Requirement Package' ontological elements in the System Engineering Subontology are used as support. A *UseCase* in OASys has been defined as a mean to capture a requirement of a system, as defined in the Requirement Package. To define the *UseCase*, the *Subject* as system under consideration, and the different *UseCaseActors*



as objects that interact with the system are also identified, among other aspects. As a result, a *System UseCase Model* is obtained, detailing the previous identified elements. The UML classes in the model are instantiations of the original OASys concepts, this fact being specified by the UML roles names in the shown associations.

When the former ontological concepts are particularised for the Robot Control Testbed, an RCT *UseCase Model* is obtained, which shows the RCT's requirements by means of use cases. A system's requirements can be of different types (physical, functional, performance, interface, design) as defined in the Requirement Package. An initial requirement analysis made by the RCT developers, identified the *FunctionalRequirements* for the RCT. A *FunctionalRequirement* is defined in OASys as a requirement that specifies an operation or behaviour that a system must perform. Primary *FunctionalRequirements* for the RCT are to navigate, as well as to survive.

The navigation *Requirement* is captured by means of the *UseCase Navigation*, which includes the secondary *FunctionalRequirements* of being able to explore the environment, to identify elements in the environment, and to avoid obstacles. These requirements are captured in the *UseCases* of EnvironmentExploration, Identification and ObstacleAvoidance respectively (Figure 10). In turn, the *FunctionalRequirement* of surviving is captured in the Survival *UseCase*, which includes the SubsystemFailure and Recharge *UseCases* (Figure 11).

It was found interesting to detail a particular *UseCase* by paying attention to the *Subsystems* in the *System*, to detail further *Requirements*. *Subsystems* identified in the RCT are the BasePlatform, the OnboardSystem, and the SupportingSystem. Focusing, for example, on the Navigation *UseCase* previously defined, it is possible to identify additional *Requirements* for the *Subsystems*, in the form of a RCT Subsystem *UseCaseModel* (Figure 12).

### C. The Process Control Testbed

The second case study to which the ontological framework is being applied is the Process Control Testbed (PCT), which is the chemical pilot plant designed to test the application of autonomous system ideas to continuous processes. Its main component is a Continuous Stirred Tank Reactor (CSTR), as well as the related instrumentation and control systems. The aim is to provide the plant with cognitive capabilities to carry out complex tasks such as fault diagnosis, alarm management, and control system reconfiguration.

The current system is shown in Figure 13. It is composed of a CSTR type reactor (R01) where two input streams are fed by their corresponding pumps (P01 and P02), and an outlet stream. The chemical reactions taking place in the reactor should be cooled with water or heated with steam, depending on the reaction. The stirring unit assures the homogeneous composition in the reactor. There is also a

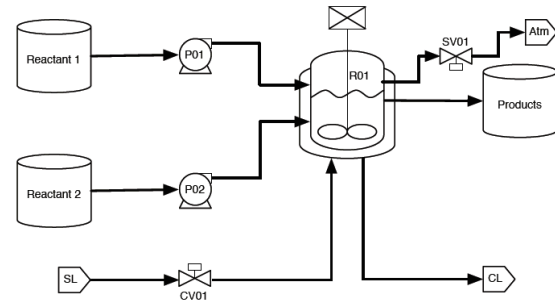


Figure 13. PCT Process Diagram

relief valve (SV01). The hydraulic subsystem is composed of the different tubing (for the reactants, the products, and the cooling water and steam), the pumps used to feed the reactants to the reactor, and the control (CV1) and relief (SV01) valves. Two small tanks for reactants storage and one for the product are also part of the system.

The instrumentation subsystem consists of several sensors (one for pH, one for temperature, and one for pressure). Moreover, there is an oxidation-reduction potential (ORP) sensor, an electrochemical analyzer, and two electromagnetic flowmeters to test and to control the different elements (reactants, products and steam) in the system. The control system is composed of a conventional computer, together with National Instrument's data acquisition boards for input/output analog and digital signals (current and voltage).

### D. PCT Conceptual Modelling

The examples of applying the OASys-based Engineering Methodology here focus on the *Structural Analysis* for the PCT, considering the *StructureViewpoint*. This *Viewpoint* pays attention to the PCT structure, as considering the *ASysStructure Concern*. A *PCTStructuralViewpointModel* can be obtained (Figure 14) by instantiating the different concepts, relationships and attributes in the *Perspective* and the *ASysPerspective Packages*. The original concepts have been ontologically instantiated into PCT related ones.

The *Structural Analysis* has as objective the analysis of a system considering its structural aspects, under a *StructuralViewpoint*. Different *Engineering Models* in the form of *Structural Models* can be obtained as result of performing the *System Modelling Subtask* defined in the *ASys Engineering Package*. The *Structural Model* is a model kind to describe an autonomous system from a structural viewpoint that conforms, as a matter of fact, to a specific level of detail that could be further refined.

For the PCT, the *Structural Model* consists of different *Structure Models* to describe the elements, as well as the *Topology Models* to describe topological connections among the components in the system. The *Structure Model* specialises the ontological elements in the *General Systems and Mereology Packages*, to describe the structural features of

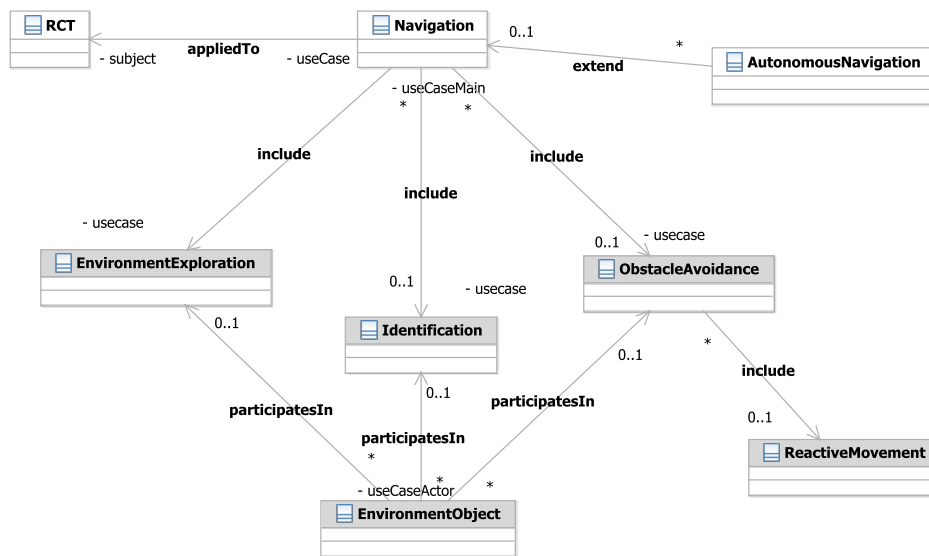


Figure 10. RCT Navigation UseCase Model

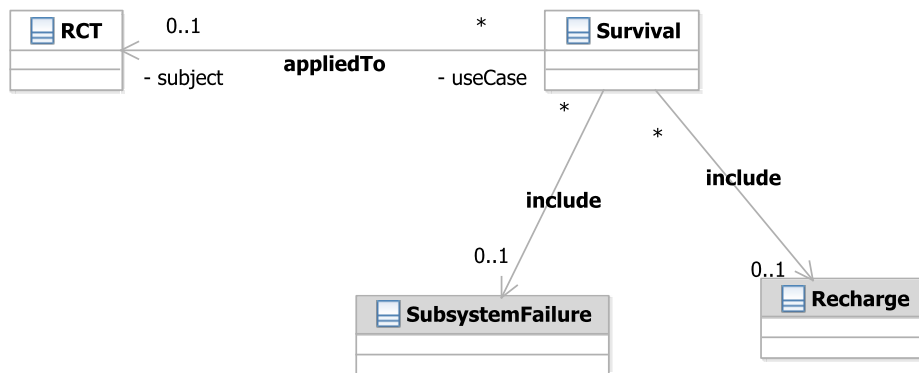


Figure 11. RCT Survival UseCase Model

the testbed. The Topology Model, not shown here, instantiates the concepts in the Topology Package.

The PCT Structure Model shows that in general, a *System* consists of *Subsystems*, i.e., a system that is constituent of another one, which in turn can be decomposed until reaching to its *Elements*, which cannot be further decomposed, as defined in the GST Package. The Process Control Testbed consists of the CSTR Reactor, the Hydraulics subsystem, the Control System, the Instrumentation, and the Power (Figure 15).

In the model, the relationships between the *System* (the PCT), and the *Subsystems* (CSTRreactor, Hydraulics, ControlSystem and Power) are expressed by means of the UML composition association. To point out that the relationships

between the different *Subsystems* and *Elements* are modelled directly using UML composition, aggregation and generalization associations, detailing the classes roles in them. If required, the associations could be changed into directed associations detailed by using the terms in the Mereology Package. However, it has not been done here to avoid cluttering the model.

Additional *Structure Models* were obtained when a particular *Subsystem* such as the CSTR Reactor (Figure 16), the Hydraulics (Figure 17), or the Instrumentation (Figure 18) were analysed.

The combined use of these ontological concepts provides for the description of the structural relations among the components in the Process Control Testbed.

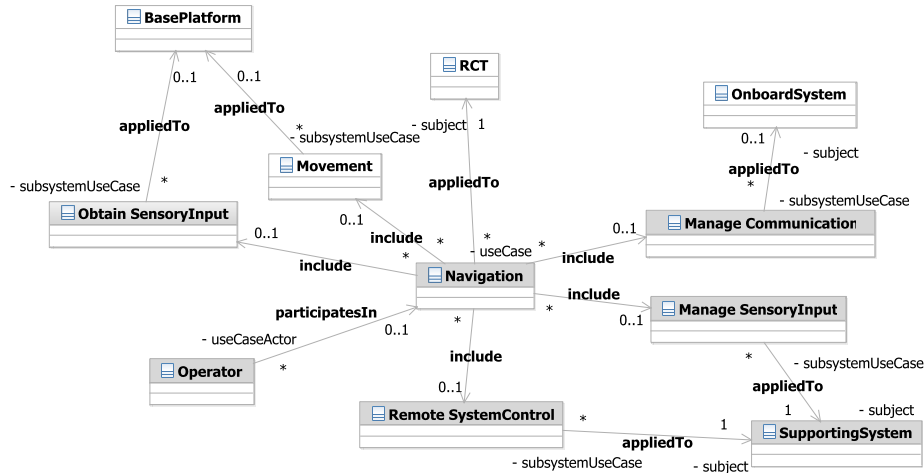


Figure 12. RCT Navigation UseCase Model detailed for Subsystems

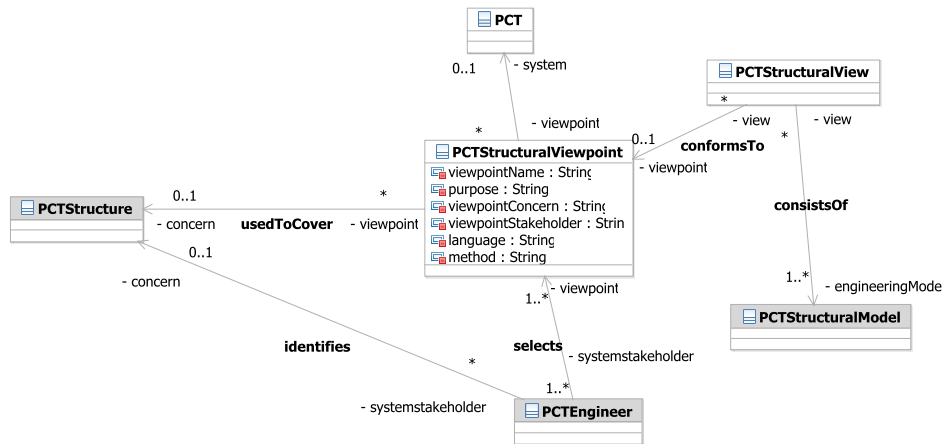


Figure 14. PCT StructuralViewpoint Model

## VI. CONCLUSION AND FURTHER WORK

Our research has considered the autonomous systems domain with a global view. Previous ontologies developed for this kind of systems focused on a constrained or limited approach, either being mobile robots or agent based systems. The approach here has been to define the different elements to describe the system structure and function in a way general enough to be reused among different applications. The resulting ontology is intended to be generically applicable to the engineering of varied systems, being this is reflected in its structure in terms of subontologies and packages.

Engineers developing autonomous systems describe and characterise them considering different elements that take part in their operation: the perception process, the knowledge to be used, the system’s goals, the functional decomposition, as well as the actors to carry out the system’s actions. OASys has catered for all these aspects, by means of the subontolo-

gies and packages that have gathered, conceptualised and formalised the ontological elements related to each one of them. OASys has come to cover the problem of modelling in a modular and unified view the complex systemic structures that many autonomous systems exhibit.

Its structure and content have allowed to scale and manage the broad range of concepts, and prevent imprecise definitions and mismatches when referring to autonomous systems. The separation between the description and the engineering aspects in a system, by means of the ASys Ontology and the ASys Engineering Ontology, has allowed to address independently the characterisation of the testbeds.

Moreover, the existence of different levels of abstraction within the ontology has shown its suitability to describe an autonomous system using a domain focalisation technique. OASys can be further complemented with additional sub-domain, task or application ontologies, without losing its reusability and generality features.

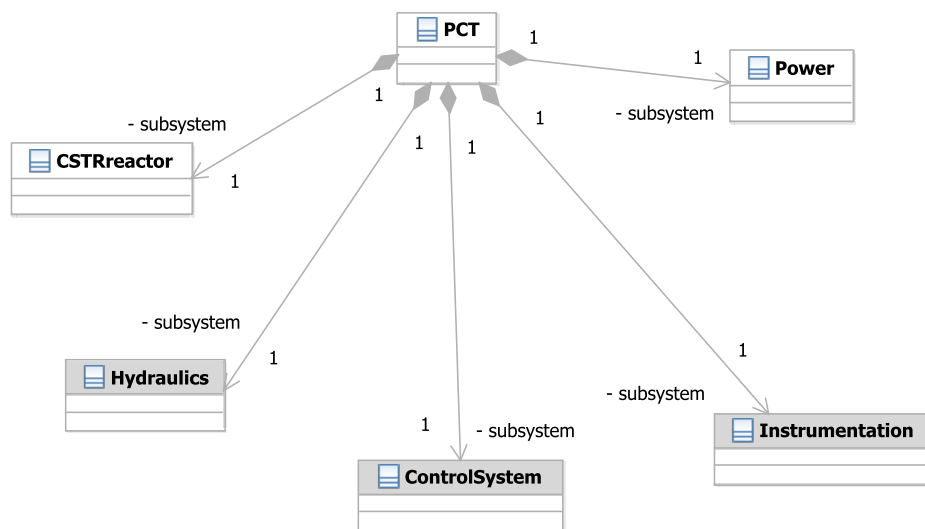


Figure 15. PCT Structure Model

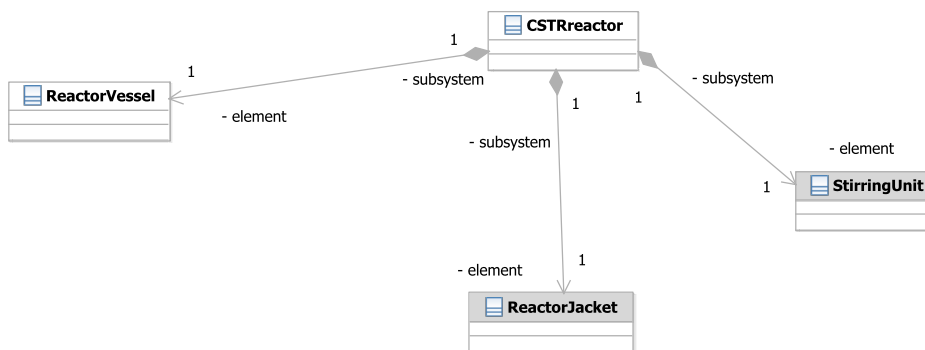


Figure 16. CSTR Reactor Structure Model

The ontology for autonomous systems has also served as an ontological metamodel to support the OASys-based Engineering Methodology. This methodology has suggested how autonomous systems can be characterised and should be engineered using conceptual models. The methodology has provided a practical approach to reuse and to extend the ontology for concrete applications. Viewpoints provide an adequate concept to tackle the complexity that is typical to this kind of systems.

The engineering methodology has provided a way of systematically devising an ontology-based account of systems going down to functional aspects that end up with the algorithmic design.

The ontology-based conceptual modelling and engineering process offers improvements on the knowledge sharing and reuse between the applications developers. OASys and its related methodology have helped to increase the

reliability of the software models to be obtained. There are no previous attempts to engineer autonomous systems ontologies and an associated methodology for its application.

The OASys-based Engineering Methodology benefits from the underlying ontological commitments and relationships to build up the autonomous system's different conceptual and engineering models, ensuring against traditional meaning and conceptual mismatches during its development.

Our aim is for any autonomous system to use the conceptual models based upon the ontological framework described in this paper, as part of a model-based systems engineering strategy [34].

An autonomous system is to perform using models of its environment, of its actions and of itself to operate, as in the model-based control paradigm, where those models will be the same ones used by the engineers to build the system. This will ultimately provide the system with self-engineering

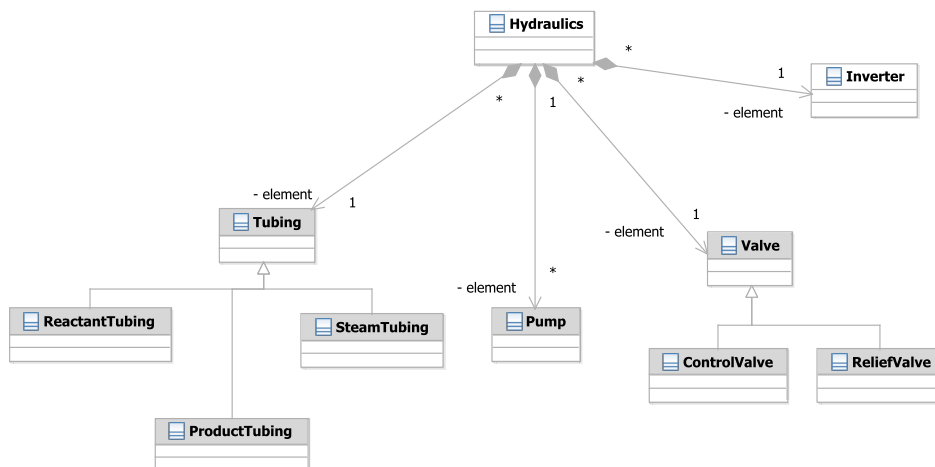


Figure 17. Hydraulics Structure Model

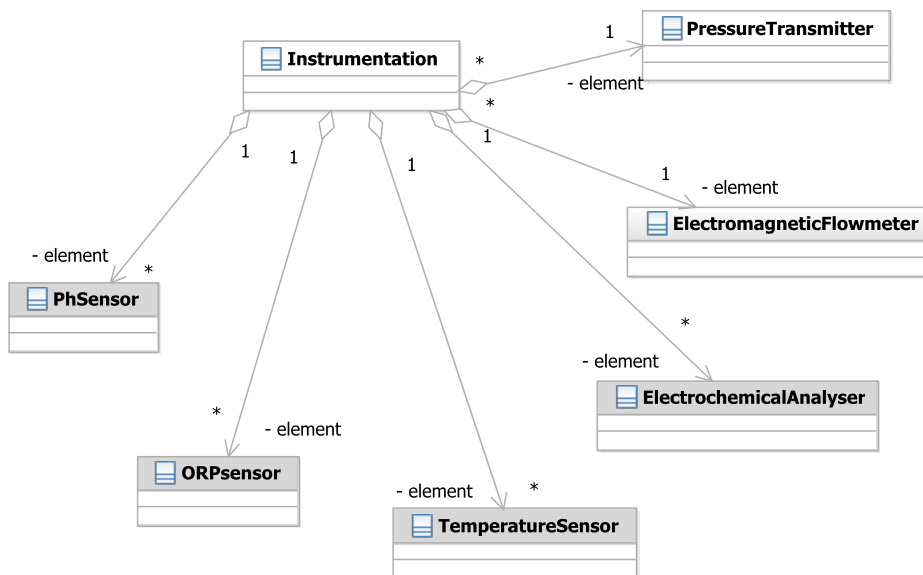


Figure 18. Instrumentation Structure Model

capabilities required for robust autonomy [35]. A further step will be for the agents to interpret those models based on OASys. It is yet to investigate how OASys and these models obtained by instantiating its ontological elements, can be used to generate meaning for the decision-making process of the different actors of an autonomous system [36].

Next stages envision the development of a methodology based on ontological and software patterns with the aid of conceptual modelling tools. This refinement process of concepts to address higher level of detail, as well as the ontological elements usage is to be defined in the ASys Modelling Methodology, as a next step in the overall ASys

research programme. This methodology will define, among other aspects, how a concept is selected, how to integrate a concept into a pattern, how to establish and to import its relationships with other concepts, and how to detail or to add its attributes in the development of a concrete model.

#### ACKNOWLEDGMENT

The authors would like to acknowledge the support of the Spanish Ministry of Education and Science (grant C3: Control Consciente Cognitivo) and the European Commission (Grant ICEA: Integrating Cognition, Emotion and Autonomy).

## REFERENCES

- [1] J. Bermejo-Alonso, R. Sanz, M. Rodríguez, and C. Hernández, "Ontology-based engineering of autonomous systems," in *Proceedings of the The Sixth International Conference on Automatic and Autonomous Systems (ICAS 2010)*, M. Bauer, J. L. Mauri, and O. Dini, Eds. Cancun, Mexico: IEEE Computer Society, 7–13 March 2010, pp. 47–51.
- [2] R. Sanz, I. Alarcón, I. Segarra, M. de Antonio, and J. Clavijo, "Progressive domain focalization in intelligent control systems," *Control Engineering Practice*, vol. 7, no. 5, pp. 665–671, 1999.
- [3] R. Sanz, C. Hernández, and M. Rodriguez, "The epistemic control loop," in *Proceedings of CogSys 2010 - 4th International Conference on Cognitive Systems*, Zurich, Switzerland, January 2010.
- [4] J. Bermejo-Alonso, "OASys: ontology for autonomous systems," Ph.D. dissertation, E.T.S.I.I.M., Universidad Politécnica de Madrid, 2010.
- [5] R. Sanz and J. Zalewski, "Pattern-based control systems engineering," *IEEE Control Systems Magazine*, vol. 23, no. 3, pp. 43–60, June 2003.
- [6] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerland, and M. Stal, *Pattern-oriented Software Architecture*. Wiley, 1996, vol. 1: a system of patterns.
- [7] T. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," in *International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation*, N. Guarino and R. Poli, Eds. Padova, Italy: Kluwer Academic Publishers, 1993.
- [8] R. Studer, V. Benjamins, and D. Fensel, "Knowledge engineering: Principles and methods," *IEEE Transactions on Data and Knowledge Engineering*, vol. 25, no. 1-2, pp. 161–197, 1998.
- [9] C. Schlenoff, S. Balakirsky, M. Uschold, R. Provine, and S. Smith, "Using ontologies to aid navigation planning in autonomous vehicles," *The Knowledge Engineering Review*, vol. 18, no. 3, pp. 243–255, 2003.
- [10] C. Scrapper and S. Balakirsky, "Knowledge representation for on-road driving," in *Proceedings of the 2004 AAAI Spring Symposium on Knowledge Representation and Ontologies for Autonomous Systems*, Stanford, California, March 2004.
- [11] C. Scrapper, S. Balakirsky, and E. Messina, "Self awareness in the mobility open architecture simulation and tools framework," in *Proceedings of the 2005 ACM workshop on Research in knowledge representation for autonomous systems*. ACM Press, 2005, pp. 35–41.
- [12] M. Uschold, R. Provine, S. Smith, C. Schlenoff, and S. Balakirsky, "Ontologies for world modeling in autonomous vehicles," in *18th International Joint Conference on Artificial Intelligence, IJCAI'03*, 2003.
- [13] T. Wagner, U. Visser, and O. Herzog, "Egocentric qualitative spatial knowledge representation for physical robots," *Robotics and Autonomous Systems*, vol. 49, pp. 25–42, 2004.
- [14] R. Provine, M. Uschold, and S. Smith, "Observations on the use of ontologies for autonomous vehicle navigation planning," in *Proceedings of the 2004 AAAI Spring Symposium on Knowledge Representation and Ontologies for Autonomous Systems*, Stanford, California, March 2004.
- [15] T. Barbera, J. Albus, E. Messina, C. Schlenoff, and J. Horst, "How task analysis can be used to derive and organize the knowledge for the control of autonomous vehicles," *Robotics and Autonomous Systems*, vol. 49, pp. 67–78, 2004.
- [16] S. Wood, "Representation and purposeful autonomous agents," *Robotics and Autonomous Systems*, vol. 49, pp. 79–90, 2004.
- [17] S. L. Epstein, "Metaknowledge for autonomous systems," in *Proceedings of the 2004 AAAI Spring Symposium on Knowledge Representation and Ontologies for Autonomous Systems*, Stanford, Palo Alto, Ca., March 2004.
- [18] H. Jung, J. Bradshaw, S. Kulkarni, M. Breedy, L. Bunch, P. Feltoovich, R. Jeffers, M. Johnson, J. Lott, N. Suri, W. Taysom, G. Tonti, and A. Uszok, "An ontology-based representation for policy-governed adjustable autonomy," in *Proceedings of the 2004 AAAI Spring Symposium on Knowledge Representation and Ontologies for Autonomous Systems*, Stanford, California, March 2004.
- [19] V. Tamma, S. Cranefield, T. Finin, and S. Willmott, Eds., *Ontologies for Agents: Theory and Experiences*, ser. Whitestein Series in Software Agent Technologies and Autonomic Computing. Birkhäuser, 2005.
- [20] Q. Chen and U. Dayal, "Multi-agent cooperative transactions for e-commerce," in *Conference on Cooperative Information Systems*, 2000, pp. 311–322.
- [21] D. Dou, D. McDermott, and P. Qi, "Ontology translation by ontology merging and automated reasoning," in *Ontologies for Agents: Theory and Experiences*, ser. Whitestein Series in Software Agent Technologies, V. Tamma, S. Cranefield, T. Finin, and S. Willmott, Eds. Birkhäuser, 2005, pp. 73–94.
- [22] A. Malucelli, D. Palzer, and E. Oliveira, "Combining ontologies and agents to help in solving the heterogeneous problem in e-commerce negotiations," in *International Workshop on Data Engineering Issues in E-Commerce (DEEC 2005)*, IEEE Computer Society, Tokyo, Japan, April 2005, pp. 26–35.
- [23] M. Nodine and J. Fowler, "On the impact of ontological commitments," in *Ontologies for Agents: Theory and Experiences*, ser. Whitestein Series in Software Agent Technologies, V. Tamma, S. Cranefield, T. Finin, and S. Willmott, Eds. Birkhäuser, 2005, pp. 19–42.
- [24] V. Tamma, "An ontology model supporting multiple ontologies for knowledge sharing," PhD, University of Liverpool, 2001.

- [25] M. Cebulla, "Knowledge-based assessment of behaviour in dynamic environments," in *Proceedings of the 2005 ACM workshop on Research in knowledge representation for autonomous systems*. Bremen, Germany: ACM Press, November 2005, pp. 17–26.
- [26] L. Stojanovic, A. Abecker, N. Stojanovic, and R. Studer, "Ontology-based correlation engines," in *Proceedings of the International Conference on Autonomic Computing (ICAC'04)*, I. Computer, Ed., 2004, pp. 304–305.
- [27] J. Martin-Serrano, J. Serrat, J. Strassner, G. Cox, R. Carroll, and M. O. Foghlu, "Policy-based context integration and ontologies in autonomic applications to facilitate the information interoperability in NGN," in *Proceedings of the Workshop on Hot Topics in Autonomic Computing*. Jacksonville, FL, U.S.A.: USENIX Association, 2007.
- [28] G. Tziallas and B. Theodoulidis, "Building autonomic computing systems based on ontological component models and a controller synthesis algorithm," in *Proceedings of the 14th International Workshop on Database and Expert Systems Applications (DEXA'03)*, Prague, Czech Republic, September 2003, pp. 674–680.
- [29] J. C. Strassner, N. Agoulmine, and E. Lehtihet, "FOCALE: a novel autonomic networking architecture," in *Proceedings of the First Latin American Autonomic Computing Symposium (LAACS 2006)*, 2006.
- [30] L. Stojanovic, J. Schneider, A. Maedche, S. Libischer, R. Studer, T. Lumpp, A. Abecker, G. Breiter, and J. Dinger, "The role of ontologies in autonomic computing systems," *IBM Systems Journal*, vol. 43, no. 3, pp. 598 – 616, 2004.
- [31] B. Chandrasekaran, J. Josephson, and V. Benjamins, "What are ontologies and why do we need them?" *IEEE Intelligent Systems*, vol. 14, no. 1, pp. 20–26, 1999.
- [32] M. Fernández-López, A. Gómez-Pérez, and N. Juristo, "METHONTOLOGY: from ontological art towards ontological engineering," in *AAAI'97 Spring Symposium on Ontological Engineering*, A. Farquhar, M. Grüninger, A. Gómez-Pérez, M. Uschold, and P. van der Vet, Eds., Stanford University, CA, U.S.A., 1997, pp. 33–40.
- [33] *OMG Unified Modeling Language (OMG UML) Superstructure Version 2.2*, Object Management Group, February 2009.
- [34] R. Sanz, C. Hernández, J. Gomez, J. Bermejo-Alonso, M. Rodríguez, A. Hernando, and G. Sanchez, "Systems, models and self-awareness: towards architectural models of consciousness," *International Journal of Machine Consciousness*, vol. 1, no. 2, pp. 255–279, December 2009.
- [35] R. Sanz, I. López, and C. Hernández, "Self-awareness in real-time cognitive control architectures," in *Proceedings of the AAAI Fall Symposium on Consciousness and Artificial Intelligence: Theoretical foundations and current approaches*, Washington, D.C., November 2007.
- [36] R. Sanz, J. Bermejo, I. López, and J. Gomez, *Toward Artificial Sapience: Principles and Methods for Wise Systems*. Springer London, 2008, ch. A real-time agent system perspective of meaning and sapience, pp. 61–73.